CAPÍTULO 2:

MANUAL DE USUARIO DEL SIMULADOR DE INSTRUCCIONES MIPS R-4000

Antes de comenzar a explicar como funciona el programa, vamos a enumerar las instrucciones que están implementadas en el simulador.

INSTRUCCIONES IMPLEMENTADAS

Instrucciones de Carga y Almacenamiento.

OpCode	Descripción
LB	Load Byte
LBU	Load Byte Unsigned
LH	Load Halfword
LHU	Load Halfword Unsigned
LW	Load Word
LWL	Load Word Left
LWR	Load Word Right
SB	Store Byte
SH	Store Halfword
SW	Store Word
SWL	Store Word Left
SWR	Store Word Right

Instrucciones Aritméticas (dato inmediato).

Opcode	Descripción
ADDI	Add Immediate
ADDIU	Add Immediate Unsigned
SLTI	Set on Less Than Immediate
SLTIU	Set on Less Than Immediate Unsigned
ANDI	AND Immediate
ORI	OR Immediate
XORI	Exclusive Or Immediate
LUI	Load Upper Immediate

Instrucciones Aritméticas (3 operandos, R-Type).

OpCode	Descripción
ADD	Add

ADDU	Add Unsigned
SUB	Subtract
SUBU	Subtract Unsigned
SLT	Set on Less Than
SLTU	Set on Less Than Unsigned
AND	AND
OR	OR
XOR	Exclusive OR
NOR	NOR

Instrucciones de Multiplicación y División.

OpCode	Descripción
MULT	Multiply
MULTU	Multiply Unsigned
DIV	Divide
DIVU	Divide Unsigned
MFHI	Move From HI
MTHI	Move To HI
MFLO	Move From LO
MTLO	Move To LO

Instrucciones de Saltos absolutos y relativos.

OpCode	Descripción
J	Jump
JAL	Jump And Link
JR	Jump Register
JALR	Jump And Link Register

BEQ	Branch on Equal
BNE	Branch on Not Equal
BLEZ	Branch on Less Than or Equal to Zero
BGTZ	Branch on Greater Than Zero
BLTZ	Branch on Less Than Zero
BGEZ	Branch on Greater Than or Eq. to Zero
BLTZAL	Branch on Less Than Zero And Link
BGEZAL	Brnch on Less Thn Zro And Link Likely

Instrucciones de desplazamiento de bits.

OpCode	Descripción
SLL	Shift Left Logical
SRL	Shift Right Logical
SRA	Shift Right Arithmetic
SLLV	Shift Left Logical Variable
SRLV	Shift Right Logical Variable
SRAV	Shift Right Arithmetic Variable

Extensiones a ISA: Instrucciones de carga y almacenamiento.

OpCode	Descripción
LD	Load Doubleword
LDL	Load Doubleword Left
LDR	Load Doubleword Right
LWU	Load Word Unsigned
SD	Store Doubleword
SDL	Store Doubleword Left
SDR	Store Doubleword Right

Extensiones a ISA: Instrucciones Aritméticas (dato inmediato).

OpCode	Descripción
DADDI	Doubleword Add Immediate
DADDIU	Doubleword Add Immediate Unsigned

Extensiones a ISA: Instrucciones de Multiplicación y División.

OpCode	Descripción
DMULT	Doubleword Multiply
DMULTU	Doubleword Multiply Unsigned
DDIV	Doubleword Divide
DDIVU	Doubleword Divide Unsigned

Extensiones a ISA: Instrucciones de salto relativo.

OpCode	Descripción
BEQL	Branch on Equal Likely
BNEL	Branch on Not Equal Likely
BLEZL	Branch on Less Than or Equal to Zro Likely
BGTZL	Branch on Greater Than Zero Likely
BLTZL	Branch on Less Than Zero Likely
BGEZL	Brnch on Greatr Thn or Equal to Zro Likely
BLTZALL	Branch on Less Than Zero And Link Likely
BGEZALL	Branch on Greater Than or Equal to Zero And Link Likely

Extensiones a ISA: Instrucc. Aritméticas (3 operandos, R-Type).

OpCode	Descripción
DADD	Doubleword Add
DADDU	Doubleword Add Unsigned
DSUB	Doubleword Subtract
DSUBU	Doubleword Subtract Unsigned

Extensiones a ISA: Instrucciones de desplazamiento de bits.

OpCode	Descripción
DSLL	Doubleword Shift Left Logical
DSRL	Doubleword Shift Right Logical
DSRA	Doubleword Shift Right Arithmetic
DSLLV	Doublewrd Shift Left Logical Variable
DSRLV	Doublwrd Shift Right Logical Variable
DSRAV	Doublwrd Shift Right Arithmetic Variable
DSLL32	Doubleword Shift Left Logical + 32
DSRL32	Doubleword Shift Right Logical + 32
DSRA32	Doubleword Shift Right Arithmetic + 32

Estensiones a ISA: Instrucciones de excepciones.

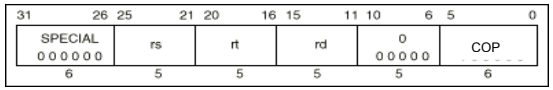
OpCode	Descripción
TGE	Trap if Greater Than or Equal
TGEU	Trap if Greater Than or Equal Unsigned
TLT	Trap if Less Than
TLTU	Trap if Less Than Unsigned
TEQ	Trap if Equal
TNE	Trap if Not Equal
TGEI	Trap if Greater Than or Equal Immediate
TGEIU	Trap if Greater Than or Equal Immediate Unsigned
TLTI	Trap if Less Than Immediate
TLTIU	Trap if Less Than Immediate Unsigned
TEQI	Trap if Equal Immediate
TNEI	Trap if Not Equal Immediate

Extensiones a ISA: Instrucciones del CP0.

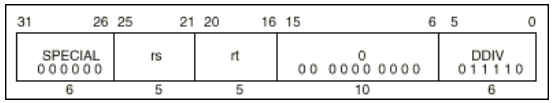
OpCode	Descripción
ERET	Exception Return

La página que viene a continuación contiene una serie de tablas que serán muy útiles a la hora de que el usuario del simulador tenga que hacer un programa en el mismo. En dichas tablas se puede ver que campos utiliza cada instrucción y el código de operación de cada una de ellas, de este modo el usuario podrá codificar fácilmente cualquier instrucción:

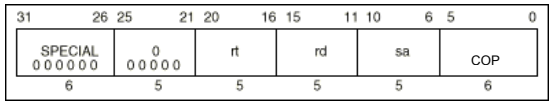
FORMATOS Y CÓDIGOS DE OPERACIÓN DE LAS INSTRUCCIONES DEL R-4000



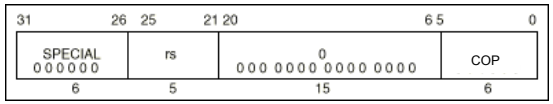
FORMATO INSTRUCCION	CÓDIGO DE OPERACION (COP)
ADD RD,RS,RT	100000
ADDU RD,RS,RT	100001
AND RD,RS,RT	100100
DADD RD,RS,RT	101100
DADDU RD,RS,RT	101101
DSLLV RD,RT,RS	010100
DSRAV RD,RT,RS	010111
DSRLV RD,RT,RS	010110
DSUB RD,RS,RT	101110
DSUBU RD,RS,RT	101111
NOR RD,RS,RT	100111
OR RD,RS,RT	100101
SLLV RD,RT,RS	000100
SLT RD,RS,RT	101010
SLTU RD,RS,RT	101011
SRAV RD,RT,RS	000111
SRLV RD,RT,RS	000110
SUB RD,RS,RT	100010
SUBU RD,RS,RT	100011
XOR RD,RS,RT	100110



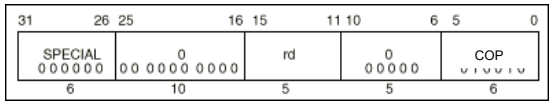
INSTRUCCION	CÓDIGO DE OPERACION (COP)
DDIV RS,RT	011110
DDIVU RS,RT	011111
DIV RS,RT	011010
DIVU RS,RT	011011
DMULT RS,RT	011100
DMULTU RS,RT	011101
MULT RS,RT	011000
MULTU RS,RT	011001
TEQ RS,RT	110100
TGE RS,RT	110000
TGEU RS,RT	110001
TLT RS,RT	110010
TLTU RS,RT	110011
TNE RS,RT	110110



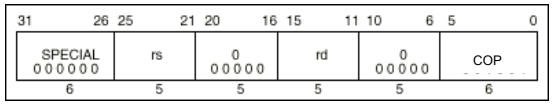
INSTRUCCION	CÓDIGO DE OPERACION (COP)
DSLL RD,RT,SA	111000
DSLL32 RD,RT,SA	111100
DSRA RD,RT,SA	111011
DSRA32 RD,RT,SA	111111
DSRL RD,RT,SA	111010
DSRL32 RD,RT,SA	111110
SLL RD,RT,SA	000000
SRA RD,RT,SA	000011
SRL RD,RT,SA	000010



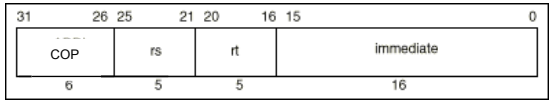
INSTRUCCION	CÓDIGO DE OPERACION (COP)
JR RS	001000
MTHI RS	010001
MTLO RS	010011



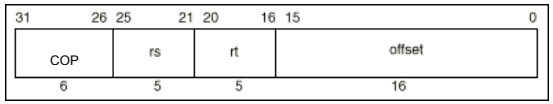
INSTRUCCION	CÓDIGO DE OPERACION (COP)
MFHI RD	010000
MFLO RD	010010



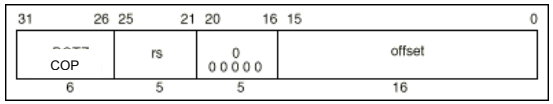
INSTRUCCION	CÓDIGO DE OPERACION (COP)
JALR RS Ó JALR RD,RS	001001



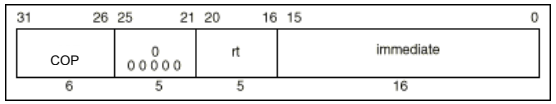
INSTRUCCION	CÓDIGO DE OPERACION (COP)
ADDI RT,RS,INMEDIATO	001000
ADDIU RT,RS,INMEDIATO	001001
ANDI RT,RS,INMEDIATO	001100
DADDI RT,RS,INMEDIATO	011000
DADDIU RT,RS,INMEDIATO	011001
ORI RT,RS,INMEDIATO	001101
SLTI RT,RS,INMEDIATO	001010
SLTIU RT,RS,INMEDIATO	001011
XORI RT,RS,INMEDIATO	001110



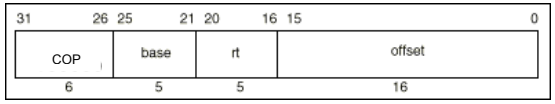
INSTRUCCION	CÓDIGO DE OPERACION (COP)
BEQ RS,RT,OFFSET	000100
BEQL RS,RT,OFFSET	010100
BNE RS,RT,OFFSET	000101
BNEL RS,RT,OFFSET	010101



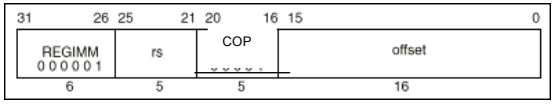
INSTRUCCION	CÓDIGO DE OPERACION (COP)
BGTZ RS,OFFSET	000111
BGTZL RS,OFFSET	010111
BLEZ RS,OFFSET	000110
BLEZL RS,OFFSET	010110



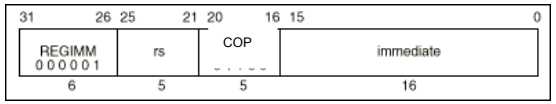
INSTRUCCION	CÓDIGO DE OPERACION (COP)
LUI RT,INMEDIATO	001111



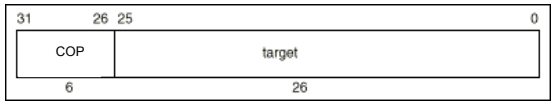
INSTRUCCION	CÓDIGO DE OPERACION (COP)
LB RT,OFFSET(BASE)	100000
LBU RT,OFFSET(BASE)	100100
LD RT,OFFSET(BASE)	110111
LDL RT,OFFSET(BASE)	011010
LDR RT,OFFSET(BASE)	011011
LH RT,OFFSET(BASE)	100001
LHU RT,OFFSET(BASE)	100101
LW RT,OFFSET(BASE)	100011
LWL RT,OFFSET(BASE)	100010
LWR RT,OFFSET(BASE)	100110
LWU RT,OFFSET(BASE)	100111
SB RT,OFFSET(BASE)	101000
SD RT,OFFSET(BASE)	111111
SDL RT,OFFSET(BASE)	101100
SDR RT,OFFSET(BASE)	101101
SH RT,OFFSET(BASE)	101001
SW RT,OFFSET(BASE)	101011
SWL RT,OFFSET(BASE)	101010
SWR RT,OFFSET(BASE)	101110



INSTRUCCION	CÓDIGO DE OPERACION (COP)
BGEZ RS,OFFSET	00001
BGEZAL RS,OFFSET	10001
BGEZALL RS,OFFSET	10011
BGEZL RS,OFFSET	00011
BLTZ RS,OFFSET	00000
BLTZAL RS,OFFSET	10000
BLTZALL RS,OFFSET	10010
BLTZL RS,OFFSET	00010



INSTRUCCION	CÓDIGO DE OPERACION (COP)
TEQI RS,INMEDIATO	01100
TGEI RS,INMEDIATO	01000
TGEIU RS,INMEDIATO	01001
TLTI RS,INMEDIATO	01010
TLTIU RS,INMEDIATO	01011
TNEI RS,INMEDIATO	01110



INSTRUCCION	CÓDIGO DE OPERACION (COP)
J TARGET	000010
JAL TARGET	000011

ERET → Código de la instrucción : 42000018 (código a introducir en el simulador).

EXPLICACIÓN DE LAS OPCIONES DEL MENÚ PRINCIPAL

A partir de aquí se van a explicar todas y cada una de las opciones del simulador, de manera que el usuario, al leerlo, pueda manejar el programa y moverse por el interfaz de usuario con más soltura.

El simulador de instrucciones del R-4000 sirve para ejecutar programas que utilizan el repertorio de instrucciones del propio microprocesador.

El menú principal se compone de 10 opciones que se muestran en la figura 1. Vamos a explicar que hace cada una de estas opciones, también habrá ejemplos gráficos:

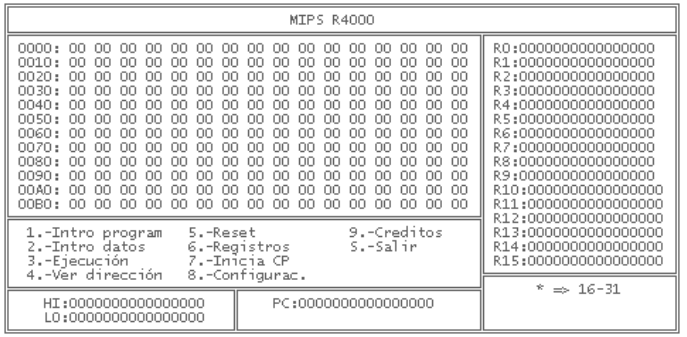


Figura 1

1.-Introducir Programa:

Esta opción nos permite introducir nuestro programa en memoria principal. Cuando pulsamos esta opción, el simulador pide una dirección de comienzo. A continuación nos pide que metamos una instrucción completa. Cuando hemos introducido la instrucción, nos pide la siguiente, y así sucesivamente hasta que metamos todas las instrucciones que forman nuestro programa. Para terminar de meter instrucciones, pulsaremos la tecla escape.

Para verlo mejor, veremos ejemplos con imágenes del propio simulador ejecutando esta opción.

Si pulsamos la opción 1 desde el menú principal, nos pide la dirección de comienzo. Esta dirección debe estar alineada en el límite de una palabra, es decir, múltiplo de 4 porque las instrucciones son de 32 bits (una palabra). Si la dirección no es múltiplo de cuatro, nos la vuelve a pedir. Introducimos la dirección 0010 y, como veremos en la figura 3, las direcciones de la pantalla se comienzan a visualizar a partir de la dirección 0010. Ya podemos comenzar a introducir instrucciones en memoria.

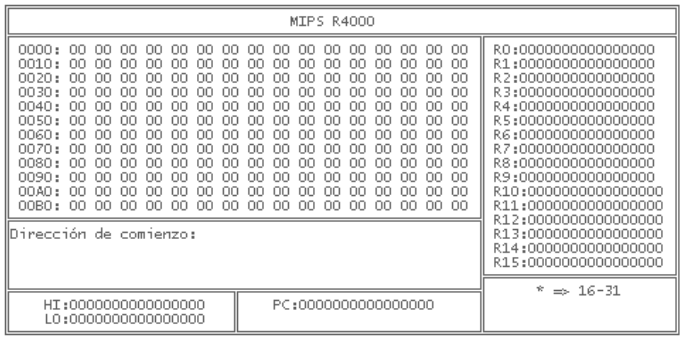


Figura 2

En la figura 3 vemos que, después de haber introducido la dirección de comienzo, las direcciones de pantalla comienzan a visualizarse a partir de la dirección 0010, entonces nos pide que introduzcamos una instrucción completa, en este caso, por hacer una prueba, vamos a meter la instrucción OR r7, r5, r3 que hace la OR lógica de los registros r5 y r3 y el resultado se guarda en el registro r7. El código que corresponde a esta instrucción concreta es 00A33825 que es el que tenemos que introducir en memoria.

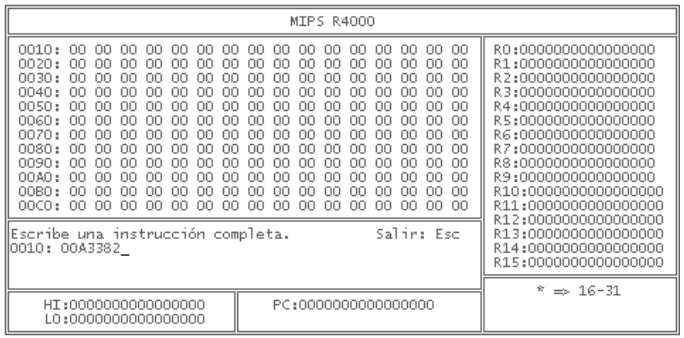


Figura 3

En la figura 4 vemos que después de escribir la instrucción, ésta se mete en la memoria y el simulador nos pide la siguiente instrucción. Las instrucciones se pueden meter en memoria de dos formas: little-endian y big-endian. La diferencia entre las dos es la ordenación de los bytes en memoria. En este ejemplo, el simulador está configurado en little-endian. Como no vamos a meter ninguna otra instrucción, pulsaremos la tecla escape para salir de la opción 1.

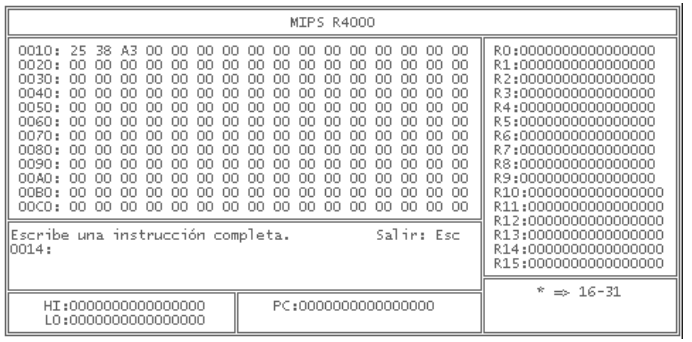


Figura 4

2.-Introducir datos

Esta opción sirve para introducir datos en memoria byte a byte. Cuando pulsamos esta opción nos piden la dirección a modificar según se muestra en la figura 5.

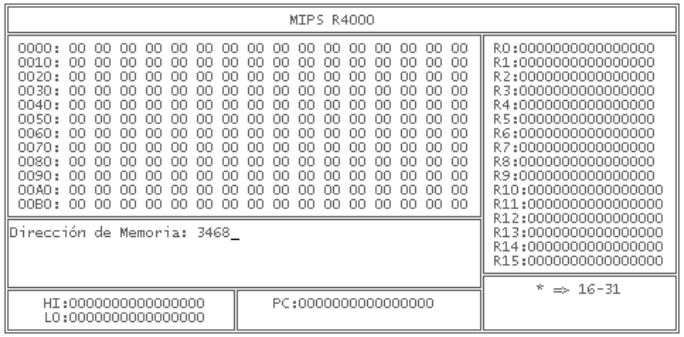


Figura 5

Cuando introducimos esta dirección, a continuación se muestra por pantalla la memoria a partir de esa dirección y pide el byte a introducir (figura 6).

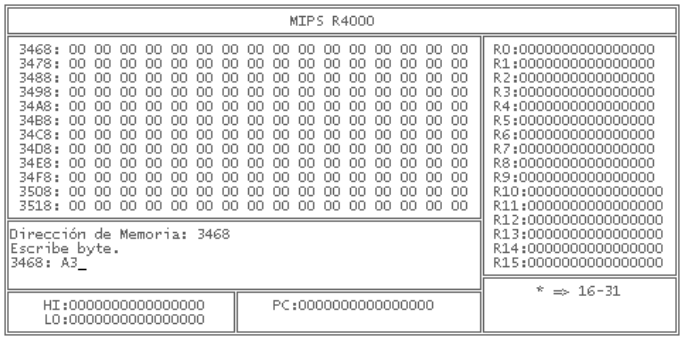


Figura 6

Metemos el byte A3 (por ejemplo). Mete el dato en la dirección especificada y nos pregunta si queremos meter otro dato en la siguiente dirección, en otra dirección distinta o si queremos terminar. Solo tenemos que pulsar una de las tres opciones. Si pulsamos la primera solo tenemos que introducir el byte que nos piden que es el de la siguiente dirección. Si pulsamos la segunda tenemos que introducir la dirección y luego el byte y si pulsamos la tercera, volvemos al menú principal. La figura 7 muestra lo explicado hasta este punto.

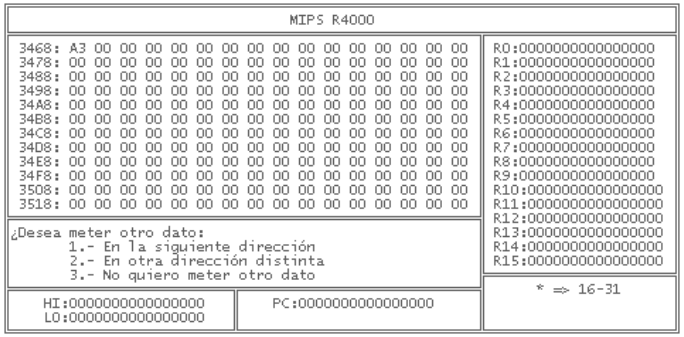


Figura 7

3.-Ejecución

Esta opción sirve para ejecutar nuestro programa a partir de la dirección a la que apunta el CP. Esta dirección se puede modificar con la opción 7 del menú principal.

Si en el menú principal pulsamos la opción 3, aparece un submenú en el que muestra al usuario las tres opciones que tiene para ejecutar un programa: Secuencialmente, en paralelo y poniendo un punto de parada (breakpoint). En la figura 8 se puede ver el submenú con las tres opciones de ejecución:

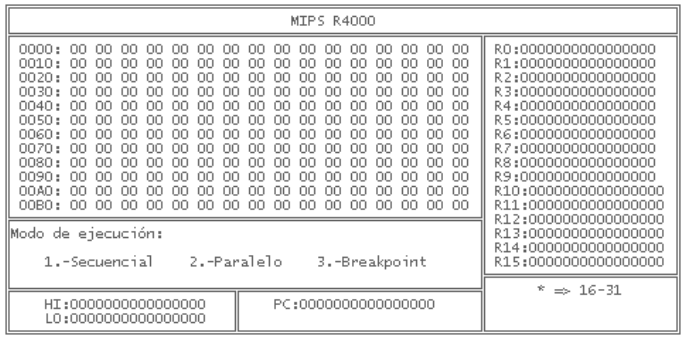


Figura 8

Si pulsamos Secuencial (opción 1), las instrucciones se ejecutarán de una en una, es decir, sale el mnemónico por pantalla; si se pulsa 'E', la instrucción será ejecutada y pasará a la siguiente. Si se pulsa 'P' la instrucción no se ejecutará y se sale al menú principal. También se puede pulsar '*' para ver el contenido de todos los registros.

Si pulsamos Paralelo (opción 2), las instrucciones se ejecutarán de dos en dos, es decir, salen por pantalla los dos mnemónicos de las dos instrucciones que se van a ejecutar; si se pulsa 'E', esas dos instrucciones se ejecutarán y pasará a las dos siguientes. Al igual que en secuencial, si se pulsa 'P' las instrucciones no se ejecutan y después de pulsar una tecla se sale al menú principal.

Por último, si pulsamos Breakpoint (opción 3), el simulador pide una dirección que será la dirección de punto de parada. Al meter la dirección, inmediatamente comienza a ejecutar instrucciones una a una hasta que llega a ese punto de parada. Después sale al menú principal.

En la figura 9 se puede ver un ejemplo de un tipo de ejecución (en este caso paralelo). Previamente hemos introducido las instrucciones ORI R3,R0,00FFh y ADDIU R4,R4,0001h. Sus códigos son respectivamente 340300FF y 24840001.

En la parte izquierda del cuadro de ejecución se puede ver cómo está configurado el simulador. Ejecución paralela (también puede ser secuencial ó breakpoint), ordenación de bytes little-endian (también puede ser big-endian), modo Usuario (también puede ser Supervisor y Kernel) y modo de ejecución 32 bits (también puede ser 64 bits).

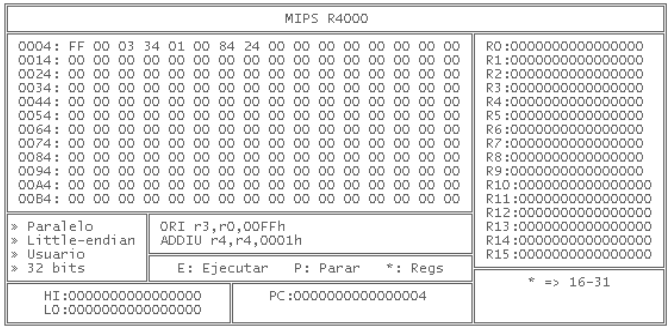


Figura 9

4.- Ver dirección

Esta opción sirve solo para visualizar la memoria a partir de una dirección dada. Al pulsar esta opción, el ordenador nos pide la dirección a partir de la cual queremos visualizar la memoria; cuando introducimos la dirección, en el cuadro donde se muestran las direcciones comienza a visualizar los bytes a partir de la dirección de memoria introducida (Ver figuras 10 y 11).

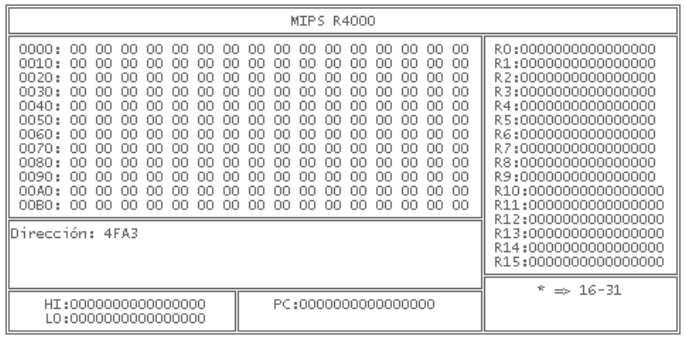


Figura 10

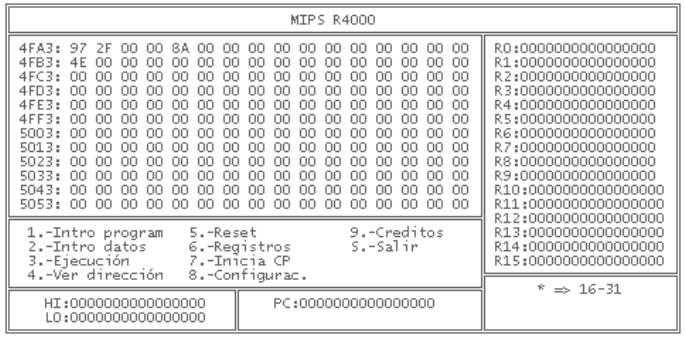


Figura 11

5.-Reset

Esta opción inicializa toda la memoria, los registros de proposito general 0..31, los de propósito especial HI, LO y el Contador de Programa (CP) a cero.

Al pulsar esta opción, el simulador pide confirmación antes de resetear el simulador según podemos ver en la figura 12. Si la respuesta es 'S' el simulador se resetea, y si es 'N' se queda tal y como está.

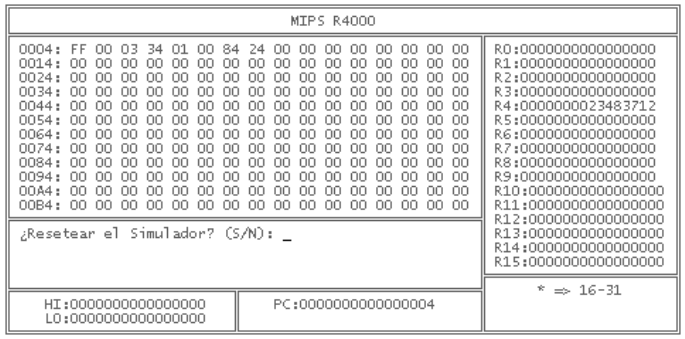


Figura 12

6.-Registros

Esta opción es para meter datos de 32 ó 64 bits en los registos de propósito general.

Como muestra la figura 13, al pulsar esta opción lo primero que nos piden es el número de registro a modificar. Solo podemos introducir un número comprendido entre 1 y 31 inclusive (R0 siempre vale 0). De no ser así, nos lo vuelve a pedir.

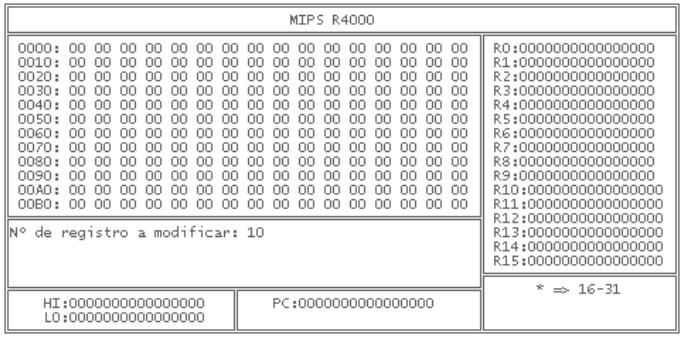


Figura 13

Cuando lo hemos introducido (en este ejemplo, registro 10), nos piden si la longitud del dato que vamos a introducir en el registro es de 32 ó de 64 bits. En el ejemplo de la figura 14 le diremos al simulador que el dato va a ser de 64 bits.

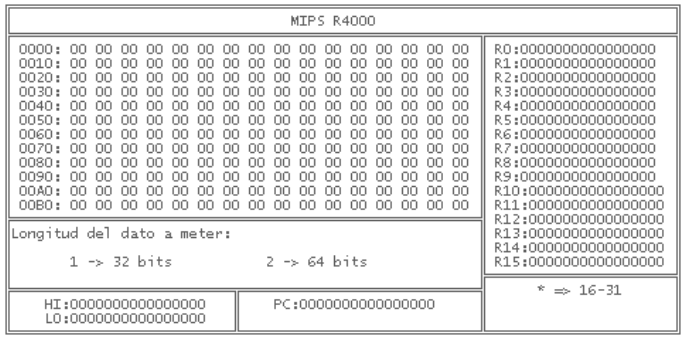


Figura 14

A continuación nos va a pedir los 32 bits de mayor peso del dato de 64 bits. Una vez metido, nos pedirá los siguientes 32 bits, que serán los de menor peso (Ver figura 15). Después de introducir todo el dato, este se guarda en el registro indicado (en este caso 10) y pregunta si se desea modificar otro registro. Si se hubiese elegido introducir un dato de 32 bits, únicamente hubiesemos tenido que introducir un dato de 32 bits. Los registros de propósito especial HI y LO, y el contador de programa (CP) no se pueden modificar desde aquí. El único modo de modificar los registros HI y LO es a través de las instrucciones de multiplicación, división y movimiento de datos.

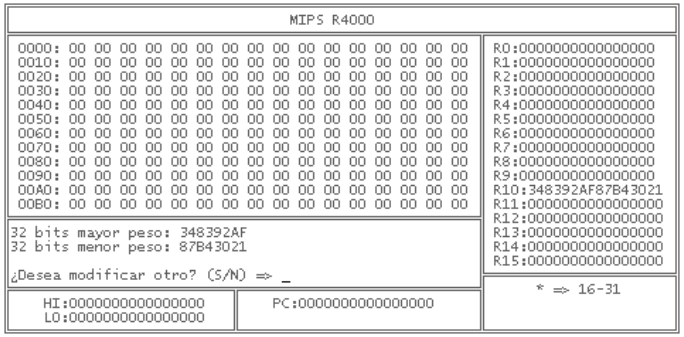


Figura 15

7.- Inicia_CP

Esta opción se suele utilizar cuando se desea que el registro CP apunte a otra dirección distinta (siempre alineada como palabra). De este modo, podemos comenzar a ejecutar un programa a partir de la dirección que nosotros queramos.

Si el valor de la dirección introducida no es múltiplo de cuatro (no está alineada como una palabra), nos volverá a pedir la dirección hasta que sea múltiplo de cuatro

8.- Configuración

Esta opción sirve para cambiar la configuración del simulador. Cuando se pulsa esta opción, el simulador muestra la configuración actual y según nos muestra la figura 16, nos pide si queremos cambiar la configuración.

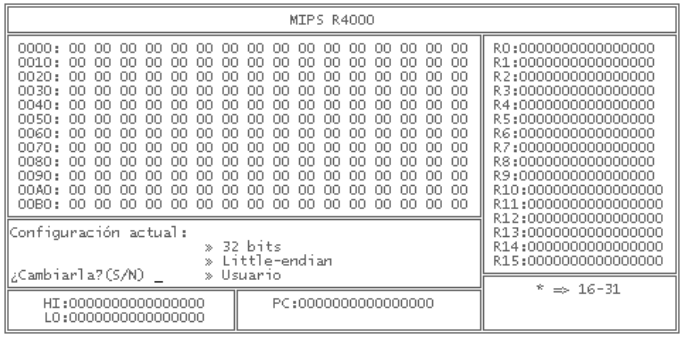


Figura 16

Si la opción pulsada es ‘N’, la configuración se queda tal y como está. Por el contrario, si la opción pulsada es ‘S’, nos da a elegir entre modo de ejecución 32 bit y modo de ejecución en 64 bits. Después da a elegir la ordenación de los bytes en memoria (little-endian ó big-endian). Ya se ha cambiado la configuración. Muestra la actual y pulsando una tecla sale al menú principal. La figura 17 muestra cómo pide el modo de operación (32 ó 64 bits).

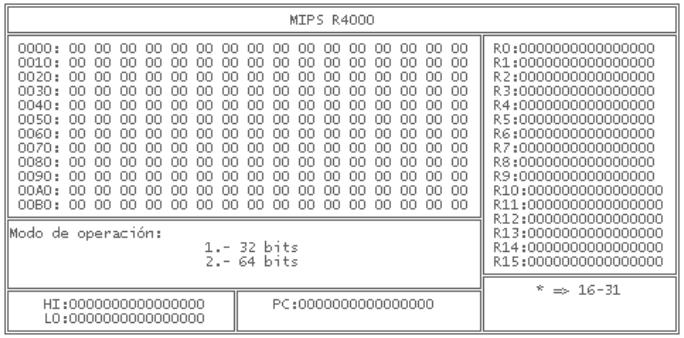


Figura 17

9.- Créditos

Como se ve en la figura 18, la única función que realiza esta opción es mostrar los datos del creador de este proyecto. Los datos que figuran son nombre, apellidos, carrera y año de finalización de este proyecto.

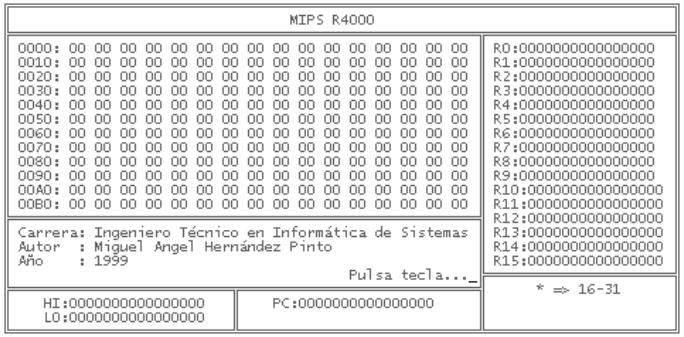


Figura 18

S.- Salir

Se sale del simulador de instrucciones del MIPS R-4000.

CODIFICAR UNA INSTRUCCIÓN

Para introducir cualquiera de estas instrucciones en el simulador, primeramente se debe hacer una codificación de la instrucción. Cada instrucción ocupa 32 bits, y dependiendo de la que sea, se necesitarán unos operandos u otros. Del mismo modo, cada instrucción tiene un código de operación; en algunas ocupa seis bits y en otras cinco bits. Si por ejemplo tenemos la instrucción **ADD r4, r5, r6** (suma con signo los contenidos de los registros r5 y r6 y el resultado lo guarda en el registro r4), y queremos obtener el código de esta instrucción (que será el que meteremos en el simulador), lo primero que debemos hacer es acudir a las tablas proporcionadas en el manual de usuario, donde está el formato, los campos y el código de operación de cada instrucción. Con esta información podremos construir el código de la instrucción arriba indicada. Si accedemos a estas tablas y buscamos la instrucción ADD, vemos que el formato es ADD rd, rs, rt. Tenemos seis campos que debemos rellenar: especial, rs, rt, rd, 0, cop. En el campo especial que ocupa 6 bits metemos seis ceros. En el campo rs queremos un cinco, por tanto metemos un cinco en binario. En el campo rt queremos un seis, por tanto meteremos un seis en binario; y en el campo rd queremos un cuatro, por tanto meteremos un cuatro en binario. El campo 0 son cinco bits a cero y en el campo COP metemos el código de operación de la instrucción ADD que es 100000.

Special	**rs**	**rt**	**Rd**	**0**	**ADD**
000000	00101	00110	00100	00000	100000

Por tanto, si estos 32 bits los separamos de 4 en cuatro bits (para poderlo pasar a hexadecimal) nos queda lo siguiente.

0000	0000	1010	0110	0010	0000	0010	0000
0	**0**	**A**	**6**	**2**	**0**	**2**	**0**

El código de la instrucción ADD r4, r5, r6 es 00A62020 y es el que debemos de introducir en el simulador para que éste lo entienda.